Invariant Manifolds as Boundaries for Event-Driven Control Systems*

Manel Velasco
manel.velasco@upc.edu

Pau Martí
pau.marti@upc.edu

Automatic Control Department
Technical University of Catalonia
Pau Gargallo 5, 08028 Barcelona, Spain

ABSTRACT

In event-driven control approaches, control updates are triggered by event conditions that are often characterized by different types of boundaries defined in the state-space domain. In this paper we define an event-driven control approach where boundaries are manifolds characterized as invariant sets. With such a boundary, a control update will only be activated when the system trajectory intersects the boundary. And the system trajectory must intersect again the boundary to activate the next control update. For linear systems, we derive an scaling property for these boundaries that permit to regulate the accuracy of the control without altering the timing offered by the scaled boundary.

1. INTRODUCTION

In event-driven control approaches, control jobs are triggered following different mechanisms, such as the measurement method itself [1], diverse forms of level-crossing mechanisms [2, 3, 4, 5, 6], state or self-triggered mechanisms [7, 8, 9, 10, 11, 12], or Lyapunov-based mechanisms [13]. These approaches exhibit diverse properties such as reducing the number of control updates with respect to periodic sampled systems or ensuring different stability and performance guarantees. Many of these mechanisms can be modeled by defining a boundary and activating control jobs whenever the system trajectory intersects the boundary [14].

By defining a complementary approach to the previous ones, in this paper we present a preliminary study on event-driven control systems where boundaries are manifolds that enforce triggering control updates only whenever the trajectory intersects a given manifold. In terms of the theory of invariant sets [15], we impose that the manifold must be an invariant set. Figure 1 illustrates this concept for a two dimensional state-space, where the manifold is plotted with a dashed line, and possible trajectories that can appear between consecutive control updates are plotted with solid arrows. For any given trajectory, a control update occurs in the manifold, and the next control update will occur again in the manifold.

*This work was partially supported by C3DE Spanish CICYT DPI2007-61527 and by ArtistDesign NoE IST-2008-214373.

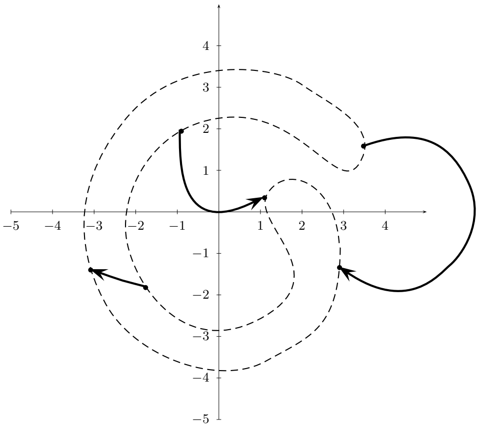

Figure 1: Manifold (dashed) and possible trajectories (arrows) that can appear between consecutive control updates

In the following, we formalize these concepts, providing several examples that illustrate the desired behavior that we define for our event-driven control approach. Then, for linear systems, we show that the timing offered by these manifolds is not altered when the manifold is scaled. Hence, the accuracy of the control can be adjusted to meet different performance specifications by appropriately scaling the manifold while not altering the timing of the control updates given by the original manifold. Finally, we discuss future work, reviewing open questions.

2. EVENT-DRIVEN CONTROL SYSTEM MODEL

We consider the continuous control system

$$\dot{x}(t) = f(t, x(t), u(t)) \tag{1}$$

where $x(t) \in \mathbb{R}^n$ denotes the state and $u(t) \in \mathbb{R}^m$ the input, respectively, at time $t \in \mathbb{R}_+$.

Let

$$\forall t \in [t_i, t_{i+1}) \quad u(t) = k(x(t_i)) = k(x_i) \tag{2}$$

be the control updates given by a feedback controller $k : \mathbb{R}^n \to \mathbb{R}^m$ using only samples of the state at discrete instants

$$t_0, t_1, \ldots, t_i, \ldots. \tag{3}$$

With (2), the closed loop system becomes

$$\dot{x}(t) = f(x(t), k(x(t_i)), t). \tag{4}$$

We denote by

$$x(x(t_i), h_i) \tag{5}$$

the solution of (4) over the time interval $h_i = t_{i+1} - t_i$ when $u(t) = k(x(t_i))$, where $x(t_i)$ is a given initial condition.

The triggering mechanism is given by a manifold characterized as invariant set for the system.

Definition 1. The manifold S is an invariant set of (4) on the control updates times t_i given in (3) if for any $x(t_i) \in S$, it holds that $x(x(t_i), h_i) \in S$. We assume that $x(t_0) \in S$, and that for any $x(t_i) \in S$ it holds that

$$\exists t_{i+1},\ t_i < t_{i+1} < \infty \mid x(x(t_i), t_{i+1} - t_i) \in S. \tag{6}$$

In the previous definition we enforce that any trajectory starting for S must return to S in a bounded time. The triggering mechanism is then defined as follows.

Definition 2. For systems (4), the triggering mechanism for control updates is defined as

$$t_{i+1} = \inf\{t > t_i \mid x(t) \in S\} \tag{7}$$

where S is given in Definition 1.

Hence, from event-driven control systems defined by (4) and(7) we can define a time function $\Lambda : \mathbb{R}^n \to \mathbb{R}$ over the boundary S that assigns to each $x(t_i) \in S$ the time interval h_i that will elapse until the trajectory will hit S again ($t_{i+1} - t_i$ in (6)), thus generating the sequence of control update times (3) for a given t_0. Then using a more compact notation, the evolution of the event-driven control system given by (4) and (7) on the control update times can be described by

$$x(t_{i+1}) = F(x(t_i), \Lambda(x(t_i))). \tag{8}$$

where $F(\cdot)$ is given by (5). Given that $x(t_i) \in S$ and $x(t_{i+1}) \in S$, an even more compact notation given by

$$S = F(S, \Lambda(S)) \tag{9}$$

will be used to stress the invariance of S.

3. EXAMPLES

In this section we present several examples that illustrate the desired operation for event-driven control systems defined by (4) and (7). Note that we do not state that they are instances of (4) and (7), but they exhibit the behavior that we are looking for. All of them use the double integrator system

$$\dot{x} = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} x + \begin{bmatrix} 0 \\ 1 \end{bmatrix} u. \tag{10}$$

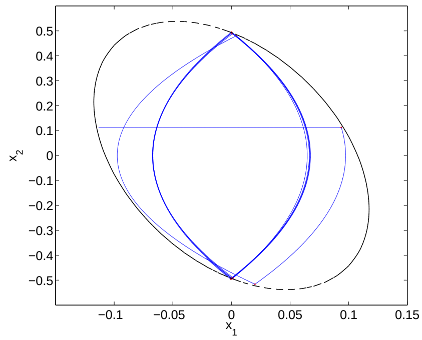

Figure 2: Event-driven dynamics achieved by a boundary based on CLFs that enforces trajectories inside S.

3.1 Example 1

Using Control-Lyapunov functions (CLF) [16] as boundaries is a clear example of the type of dynamics that we are interested on. For non-linear affine systems

$$\dot{x} = f(x) + g(x)k(x), \tag{11}$$

being S a contour curve of a Lyapunov function V, i.e. $S : V(x) = c$, if V is a CLF when evaluated in the boundary, i.e.

$$\forall x \in S \quad \text{if } \nabla V(x)g(x) = 0 \text{ then } \nabla V(x)f(x) < 0, \tag{12}$$

and the controller ensures that

$$\forall x \in S, \nabla V(x)(f(x) + k(x)g(x)) < 0, \tag{13}$$

as the universal controller [16] does, then the trajectory will always lie inside S, and control updates are only activated when the trajectory intersects S, pushing the state inside S.

Using the double integrator (10), let

$$V(x) = x^T P x \text{ with } P = \begin{bmatrix} 1.1455 & 0.1 \\ 0.1 & 0.0545 \end{bmatrix}$$

be the Lyapunov function, and let S be defined as

$$S : V(x) = 0.01.$$

With initial condition

$$x_0 = \begin{bmatrix} -0.11 \\ 0.11 \end{bmatrix}$$

Figure 2 shows the system trajectory achieved when control updates are triggered according to (7) making use of the universal controller given in (14). As it can be seen the trajectory lies inside S (dashed ellipse) as time progresses and control updates are triggered when it intersects S.

By using CLF with a universal controller it is ensured that $\dot{V}(S) < 0$. However, it is also interesting to relax this condition, and to permit $\dot{V}(S) > 0$ while still ensuring (6), meaning that the trajectory can escape from inside S but must return to S.

$$k(x) = \left(\left(-\frac{23}{10}\, x_1 - 1/5\, x_2 \right) x_2 - \sqrt{ \left(\frac{23}{10}\, x_1 + 1/5\, x_2 \right)^2 {x_2}^2 + \left(1/5\, x_1 + \frac{11}{100}\, x_2 \right)^4 } \right) \left(1/5\, x_1 + \frac{11}{100}\, x_2 \right)^{-1} \tag{14}$$

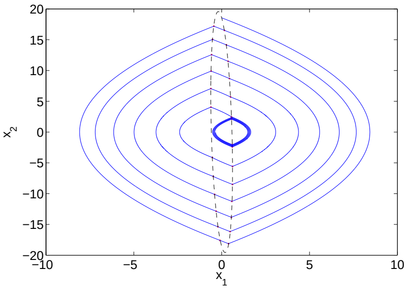

Figure 3: Event-driven dynamics achieved by an alternative boundary that permits trajectories inside and outside S.

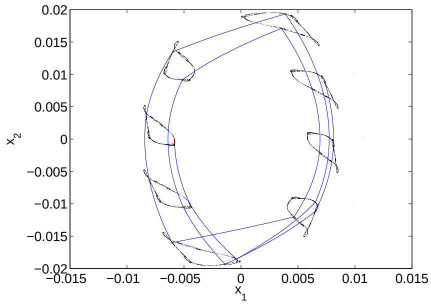

Figure 4: Event-driven dynamics for a multivalued boundary.

3.2 Example 2

To illustrate the previous observation, by using the double integrator system (10), let the Lyapunov function be

$$V(x) = x^T P x \text{ with } P = \begin{bmatrix} 10 & 0.1 \\ 0.1 & 0.01 \end{bmatrix},$$

and let the control updates be given by a linear state feedback control law with gain

$$L = \begin{bmatrix} -0.1 & -1.1 \end{bmatrix}.$$

With initial condition

$$x_0 = \begin{bmatrix} 0 \\ 18.56 \end{bmatrix}$$

Figure 3 shows the system trajectory achieved when control updates are triggered according to (7) when S (dashed ellipse) is defined as

$$S : x^T P x = x_0 P x_0.$$

As it can be seen the trajectory not always lies inside the boundary but control updates are triggered when it intersects the boundary S, thus achieving the behavior imposed to our event-driven control system approach. In this case, $\dot{V}(S) > 0$ is permitted.

3.3 Example 3

To show the complexity of these type of boundaries, this example, also using the double integrator system (10), and having control updates given by a linear state feedback control law with gain

$$L = \begin{bmatrix} -5 & -1.5 \end{bmatrix},$$

defines the boundary using an alternative approach. Rather than imposing a boundary, a time function Λ for specifying the control update times defined as

$$h_i = \sqrt{\frac{0.012}{((A+BL)x_i)^T((A+BL)x_i)}}$$

has been imposed, where A and B are the state and input matrices of (10), and x_i is each sampled state, with

$$x_0 = \begin{bmatrix} 0.0074 \\ 0.0087 \end{bmatrix}.$$

Using this function, the boundary shown in Figure 4 has been obtained numerically. It can be seen that the boundary is a multivalued function and that control updates are only triggered when the trajectory intersects it, achieving again the desired behavior.

4. MANIFOLD SCALING PROPERTIES FOR LINEAR SYSTEMS

Consider the linear instance of the event-driven control system defined by (4) and (7), that is, where (4) is given by

$$\dot{x}(t) = Ax(t) + BLx_i \tag{15}$$

where L is a linear controller gain.

Proposition 1. *Given a linear event-driven control system defined by (15) and (7), the control update times will remain the same if the manifold S in (7) is scaled by $k \in \mathbb{R}_+$.*

Proof. Using the compact notation given in (9), the proposition states that $\Lambda(\cdot)$ is invariant with respect to scalings of S. Let kS denote the manifold S scaled by k. For linear systems (15), $F(\cdot)$ in (8) applied to $x(t_i) \in S$ becomes

$$x(t_{i+1}) = F(x(t_i), h_i) = e^{Ah_i}x(t_i) + \int_0^{h_i} e^{As} ds BLx(t_i), \tag{16}$$

that applied to S or kS using (9) respectively transforms to

$$S = F(S, \Lambda(S)) = e^{A\Lambda(S)}S + \int_0^{\Lambda(S)} e^{As} ds BLS, \tag{17}$$

$$kS = F(kS, \Lambda(kS)) = e^{A\Lambda(kS)}kS + \int_0^{\Lambda(kS)} e^{As} ds BLkS. \tag{18}$$

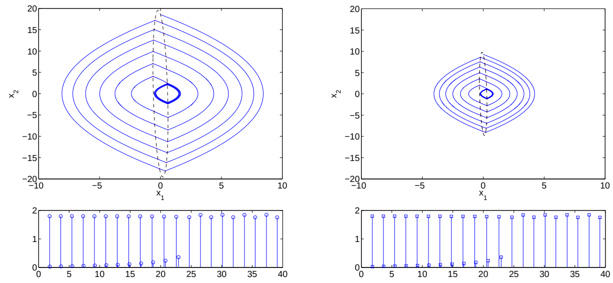

Figure 5: Dynamics (top) and timings (bottom) of an event-driven control system for a given S (left) and for a scaled S (right).

Now, applying $F(\cdot)$ to kS while keeping the same timing achieved with S, that is, $\Lambda(S)$, we obtain

$$\begin{aligned} F(kS, \Lambda(S)) &= e^{A\Lambda(S)}kS + \int_0^{\Lambda(S)} e^{As} ds BLkS \quad (19) \\ &= k(e^{A\Lambda(S)}S + \int_0^{\Lambda(S)} e^{As} ds BLS) \\ &= kS \end{aligned}$$

Hence, comparing (18) and (19), it follows that $\Lambda(S) = \Lambda(kS)$. $\square$

It must be pointed out that kS denotes the manifold S scaled by k. If S is given in parametric form, then kS directly denotes the product.

This property is important because the accuracy of the control can be increased without increasing the number of control updates, and thus, keeping the same processor or bandwidth demand in networked and embedded control systems designed using the presented event-driven control approach.

In order to illustrate the result given by Proposition 1, we recover Example 2 (section 3.2) and rescale S as

$$S : x^T P x = \frac{1}{2} x_0^T P \frac{1}{2} x_0.$$

Figure 5 shows in the left sub-figures the dynamics of the original system (top) with the sampling intervals h_i of consecutive control updates (bottom, where the x-axe is time in ms and the y-axe is h_i) while in the right sub-figures the same information is shown for the rescaled S. The dynamics are different while the time intervals between consecutive control updates are the same.

5. CONCLUSIONS AND FUTURE WORK

This paper has defined a novel approach to event-driven control systems where control updates are only triggered when the system trajectory intersects a given manifold defined as an invariant set. Many open questions arise. First of all, given a manifold S, it must be shown that it is an invariant set for the system on the control updates times. Second, given S, procedures are required to find $\Lambda(\cdot)$, or vice-versa, given $\Lambda(\cdot)$ find S (as in example 3). Note that this question is related to the first one. Other important issues are related to chattering (what happens if S is a limit cycle for the system?), to equilibrium points that may exist inside S and may prevent having infinite impacts on the S as mandated by the defined approach, or questions such as how to bring the state from "outside" S to S.

6. REFERENCES

[1] W.P.M.H. Heemels, R.J.A. Gorter, A. van Zijl, P.P.J. van den Bosch, S. Weiland, W.H.A. Hendrix and M.R. Vonder. Asynchronous measurement and control: a case study on motor synchronization. *Control Engineering Practice*, 7:1467–1482, 1999.

[2] K.-E. Årzén. A Simple event-based PID controller. In *14th World Congress of IFAC*, January, 1999.

[3] K.J. Åström and B. Bernhardsson. Comparison of Riemann and Lebesgue sampling for first order stochastic systems. In *41st IEEE Conference on Decision and Control*, December 2002.

[4] V.H. Nguyen and Y.S. Suh. Networked estimation with an area-triggered transmission method. *Sensors*, 8(2):897–909, 2008.

[5] T. Henningsson, E. Johannesson, A. Cervin. Sporadic event-based control of first-order linear stochastic systems. *Automatica*, 44(11):2890–2895, Nov. 2008.

[6] D.V. Dimarogonas and K.H. Johansson. Event-triggered control for multi-agent systems. In *48th IEEE Conference on Decision and Control*, Shanghai, China, December 2009.

[7] M. Velasco, P. Martí and J.M. Fuertes. The self-triggered task model for real-time control systems. In *WiP of the 24th IEEE Real-Time Systems Symposium*, December 2003.

[8] P. Tabuada. Event-triggered real-time scheduling of stabilizing control tasks. *IEEE Trans. on Automatic Control*, 52(9):1680–1685, 2007.

[9] M. Lemmon, T. Chantem, X. S. Hu, and M. Zyskowski. On self-triggered full information h-infinity controllers. In *Proc. of the* 10^{th} *International Conference on Hybrib Systems: Computation and Control*, Apr. 2007.

[10] A. Anta and P. Tabuada. Self-triggered stabilization of homogeneous control systems. In *2008 American Control Conference*, 2008.

[11] W.P.M.H. Heemels, J.H. Sandee, and P.P.J. van den Bosch. Analysis of event-driven controllers for linear systems. *International Journal of Control*, 81(4):571–590, 2008.

[12] X. Wang and M. Lemmon. Self-triggered feedback control systems with finite-gain L2 stability. *IEEE Trans. on Automatic Control*, 54(3):452–467, 2009.

[13] M. Velasco, P. Martí and E. Bini. On Lyapunov sampling for event-driven controllers. In *48th IEEE Conference on Decision and Control*, Shanghai, China, December 2009.

[14] M. Velasco, P. Martí and E. Bini. Control-driven tasks: modeling and analysis. In *29th IEEE Real-Time Systems Symposium*, December 2008.

[15] F. Blanchini. Set invariance in control. *Atomatica*, 35:1747–1767, 1999.

[16] E.D. Sontag. A Universal Construction of Arststein's Theorem on Nonlinear Stabilization. *System Control Letters*, 13:117–123, 1989.